IN THE CLAIMS:

1. (currently amended) A processor comprising:

an execution unit to execute a set of instructions; and

an instruction fetching mechanism that retrieves the set of instructions to be executed by the execution unit, at least one of the set of instructions comprising a single instruction that provides ~~for execution of other instructions of the set of instructions in accordance with multiple looping constructs~~ information corresponding to at least two different loops, wherein the at least two different loops have at least one of different start addresses or different end addresses.

2. (currently amended) The processor of claim 1, wherein the single instruction provides the information to initialize the at least two different loops ~~initializes a plurality of loops~~ for later execution.

3. (currently amended) The processor of claim 1, wherein the ~~multiple looping constructs~~ at least two different loops are implemented in a nested structure.

4. (currently amended) The processor of claim 1, wherein the single instruction includes ~~a plurality of loop termination conditions~~ at least one loop termination condition corresponding to at least one of the at least two different loops.

5. (currently amended) The processor of claim 1, wherein the single instruction includes at least one field that identifies a location of a last instruction of ~~a loop~~ at least one of the at least two different loops.

6. – 19. (cancelled)

20. (currently amended) A method of performing an instruction for use by a processor, the method comprising:

fetching a single instruction from a memory source, wherein the single instruction ~~provides for execution of~~ corresponds to multiple looping constructs, the multiple looping constructs including a first loop and a second loop to be executed;

decoding the single instruction; and

in response to decoding the single instruction, using information provided by the single instruction to initialize a plurality of loop storage elements corresponding to the first loop and the second loop, wherein the first loop and the second loop have at least one of different start addresses or different end addresses.

21. (previously presented) The method of claim 20, wherein the multiple looping constructs are nested such that the first loop is nested within the second loop.

22. (previously presented) The method of claim 20, wherein the information provided by the single instruction to initialize the plurality of loop storage elements includes at least one termination condition for each of the first loop and the second loop.

23. (previously presented) The method of claim 22, wherein the at least one termination condition for each of the first loop and the second loop comprises at least one of a loop count value and a condition code.

24. (previously presented) The method of claim 22, further comprising:

performing a first set of logic relating to the at least one termination condition for the first loop; and

performing a second set of logic relating to the at least one termination condition for the second loop.

25. (currently amended) The method of claim 20, wherein the information provided by the single instruction to initialize the plurality of loop storage elements includes an end of loop ~~indication~~ address for each of the first loop and the second loop.

26. (previously presented) The method of claim 20, wherein the single instruction is a first instruction of the first loop and a first instruction of the second loop, the first instruction of the first loop and the first instruction of the second loop being a same instruction at a same instruction address.

27. (previously presented) The method of claim 26, wherein the single instruction is one of a beginning instruction of the first and second loops and an ending instruction of the first and second loop.

28. (previously presented) The method of claim 20, further comprising:
determining a loop type corresponding to the single instruction, wherein the loop type is selected from one of a conditional and non-conditional type.

29. (New) The processor of claim 4, wherein the at least one loop termination condition comprises one of a loop count value or a condition code.

30. (New) The processor of claim 1, wherein the single instruction includes a loop termination condition corresponding to each of the at least two different loops.

31. (New) The processor of claim 30, wherein each loop termination condition comprises one of a loop count value or a condition code.

32. (New) The processor of claim 1, wherein the single instruction includes fields that identify a location of a last instruction of each of the at least two different loops.

33. (New) The method of claim 22, further comprising:

determining a loop type for each of the first loop and the second loop, wherein the loop type is selected from one of a conditional and non-conditional type;

if the loop type is conditional, the termination condition comprises a condition code; and

if the loop type is non-conditional, the termination condition comprises a loop count value.

34. (New) The method of claim 23, wherein the information provided by the single instruction to initialize the plurality of loop storage elements includes an end of loop address for each of the first loop and the second loop.

35. (New) The method of claim 23, wherein the information provided by the single instruction to initialize the plurality of loop storage elements includes a start of loop address for each of the first loop and the second loop.